Claims

[c1] DE920000099US1

1.A method for operating an out of order processor in which a rename process is comprised of the pipeline an instruction stream is processed with, the method comprising the steps of:

processing the pipeline in a compressed way thereby risking dependency conflicts;

providing a separate logic for detecting a dependency conflict associated with an instruction currently to be renamed;

setting a conflict flag reflecting the detection result; and

continuing the processing dependent on the conflict flag.

[c2] 2.The method according to claim 1 in which the step of continuing the processing comprises the step of setting an interrupt bit in order to reset the pipeline.

[c3] 3.The method according to claim 1 in which the step of continuing the processing comprises the step of flushing the pipeline.

[c4] 4.The method according to claim 1 in which the step of continuing the processing comprises the step to provide a reservation station (RS) with the missing information which caused the conflict, and continuing to process the same pipeline status without resetting or flushing it.

[c5] 5.The method according to claim 4 in which the step of detecting said dependency conflict comprises the steps of:

reading an instruction tag and a valid bit of a ReOrder Buffer (ROB) entry;

determining that said valid bit was modified without being tracked by the RS ;

setting said conflict bit for indicating that said entry has to be issued to the RS ;

issuing said tag to an additional port of the tag compare logic ; and

triggering the write of said valid bit into a respective field of said RS.

[c6] 6.The method according to claim 5 in which the ROB entry comprises result data currently written by an Instruction Execution Unit (IEU), further comprising the step of copying said result data into the respective entry of the RS.

[c7] 7.A processing system for operating an out of order processor in which a rename process is comprised of a pipeline having an instruction stream being processed, said processing system having means for executing a readable machine language, said readable machine language comprises:
a first computer readable code for processing the pipeline in a compressed way thereby risking dependency conflicts;
a second computer readable code for providing a separate logic for detecting a dependency conflict associated with an instruction currently to be renamed;
a third computer readable code for setting a conflict flag reflecting the detection result; and
a fourth computer readable code for continuing the processing dependent on the conflict flag.

[c8] 8.The processing system according to claim 7 in which an additional port is provided for a reservation station and a reorder buffer for detecting said conflict and continuing the processing dependent on the conflict flag.

[c9] 9.A computer system having an out of order processing system, said computer system executes a readable machine language, said readable machine language comprises:
a first computer readable code for processing the pipeline in a compressed way thereby risking dependency conflicts;
a second computer readable code for providing a separate logic for detecting a dependency conflict associated with an instruction currently to be renamed;
a third computer readable code for setting a conflict flag reflecting the detection result; and
a fourth computer readable code for continuing the processing dependent on the conflict flag.